RoboMind Course plan Primary 2

Courseplan

Overview of the RoboMind lessons

General info

Subject	**RoboMind Course plan Primary 2**
Level	For children aged 10 years or older
Prerequisites	Basic computer skills, RoboMind Primary 1. (file opening and saving, simple text editing, repeat loops, conditions)
Materials	Presentations, exercises and hints (pdf), movies (mp4), software, scripts and maps (RoboMind)
Duration	7 weeks, 1.0-1.5 hours per week
ICT Curriculum	UK: KS2,3 POS: 1.3 (a and b) 2.2 (b and c), 2.4 (a, b, and c)

Goal

The aim is to gain insight into logic, automation and robotics by programming a virtual robot with the programming language ROBO. This will also give direct insight into the operation of technical appliances as they are all around us.

ROBO is a small language with a concise set of rules where no prior knowledge is required, so students can get started straight away. As the children make the exercises, they get acquainted with the possibilities and impossibilities of the programming language and acquire more insight into the power of logic. They will also gradually learn how a large problem can be solved by breaking it up into smaller pieces which can be solved more easily.

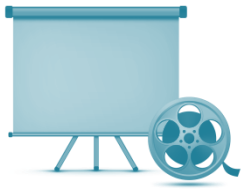
These lessons introduce *procedures* and the use of *randomness* as a way to solve automation problems. In addition *repeat* loops and *conditional statements* are rehearsed and extended. The *"if ... else ..."* construct is extended to *"if ... else if ... else"*. Conditions are extended to multiple conditions of the form *"If (A and B) ..."*. Several quizzes have been added to the presentations to train the use of such constructs and to make sure everyone really understands it. To this end there is the "if-else" quiz in presentation 1, the "if-else-if" quiz in presentation 2, and the "if-this-and-that" quiz in presentation 5.

Overall Lesson 1 and 2 repeat and extend the first set of lessons (RoboMind Course Primary School 1). Then procedures are introduced in Lesson 3, 4, and 5, each time from a different perspective. The aim is make students familiar with procedures, to show them how procedures can be used, and why and when procedures can be useful. Finally, in Lesson 6 and 7, the concept of randomness is introduced as a strategy to solve automation problems.

During the last lesson (7) the students do a kind of 'masterpiece'. The problem is not very difficult, but involves using all they learned in the previous lessons. That is: procedures and double conditions and the randomness automation strategy. The lesson is concluded with a short evaluation and a last fun robot movie.

General structure of the lessons

All lessons have the following structure.



Every lesson starts with a short multimedia presentation which highlights some aspects of what robots do en what their role is in society. In addition a brief explanation is given about the programming instructions used in the exercises and why these could be useful. This part takes about 15-20 minutes.

Next, the children will work for themselves with the virtual robot 'Robo' in RoboMind. They will be asked to let Robo do tasks as set out in the exercises. There should be no more than two children behind each computer.

It is important to let the children solve the problems mostly by themselves. Robo is a virtual robot, so you can try as often as you want, and you can never do wrong. However, the teacher can always offer new clues (provided in a separate document) if a problem proves too difficult to solve. Thus basically every problem can be solved eventually by everyone. This part takes ± 45 minutes.

The more exercises a student finishes, the better. However, it does not matter for the series of lessons if not all exercises have been completed. For quick learners, we provide additional exercises toward the end of the programming activity, such that the smart ones do not get bored.

After the programming activity a brief discussion of 5 to 10 minutes is recommended. Here the children can be given the opportunity to tell what was fun and what was considered to be (very) challenging. This can then be given some extra attention next time.

Planning over the weeks

Week	Content
Week 1	**Tracking and Tracing**. This first lesson starts with some general repetition of using repeat loops and conditions. The robot has to follow a trail of white dots in several maps to get to the beacon. This scheme is then extended to a trail with white *and* black dots where the robot has to go a different direction. The final exercises are to follow a trail which is encoded with two white dots. The encoding is a sort of secret and students are to apply double conditions of the type if (A and B) ... to solve it. This can be quite tricky and is only meant for the most advanced students.
Week 2	**Finding beacons in a maze.** This lesson is about how to find beacons in an arbitrary maze. The students learn that if you keep the wall to your right hand side you will always find the beacon sooner or later. They will program this strategy themselves during the exercises. First they are asked to write down what the robot should do in every possible situation. Then they will slowly extend their program by adding new situations (conditions) to it in every next exercise. The structure of the program is already given to a large extent as the aim is mainly to train using if ... else if ... else ... structures and the associated conditions. For the most advanced students there is a final exercise where they will have to solve a maze with obstacles in it.
Week 3	**Secret language.** In this lesson procedures are introduced for the first time. Students are asked to make procedures which encode how to write the letters for 'A', 'P', and 'L' in a sort of secret language. They are then asked to let Robo write words with it. Writing in a secret language allows the procedures to be much simpler so they can concentrate on the use of procedures rather than on the contents. Otherwise it is also more fun ☺ Writing words shows that procedures are very handy for sequencing. Otherwise it shows that you will only have to create a procedure once and from then on can reuse them any time. The overall aim is the introduction of the procedure and to show how it might be useful.
Week 4	**Drawing robot portraits.** In this lesson we will first make a robot portrait and then a whole series of portraits if there is any time left. In order to efficiently (re)use a procedure in sequences, it is important that you decide how the procedure ends: in what relative position and orientation should the robot be after the procedure has finished? It is important and necessary for the students to understand this in order to successfully solve the exercises. For the fastest there is again an additional exercise which is to create a new procedure that uses procedures itself and encodes drawing one complete robot portrait. With this procedure a whole series of robot portraits can then be drawn (a complete robot family portrait if you like).
Week 5	**Reading barcodes.** The task is to read the barcode from products on the conveyor belt in the supermarket. The barcode tells Robo what product it is and where to put it back. This is a somewhat different lesson in the sense that the whole program is already given. Only the procedure "putProductBack" is to be modified. Both the recognition of the

	barcode (double condition) and the put-back location have to be adjusted. This is done step-by-step with one new product introduced each time. The aim is to teach students how to read and understand a larger program in such a way that they are able to adjust it. At the same time they will see an example of how procedures can be used. For the fastest there is the possibility to make a mess of things in Robo's supermarket ☺
Week 6	**Vacuum cleaning.** In this lesson the notion of randomness is introduced as a strategy to solve automation problems. First randomness is explained and students are asked to make a sort of die with RoboMind. Students are asked to play dice with Robo several times and to write down how many eyes are thrown. This way they will get a little bit of insight into the concept of randomness. They will then use this to program Robo to find a beacon which is located in an arbitrary place. They will look at the number of steps it takes to find the beacon if they repeatedly run the program on the same map and see if this is always the same (no). Finally they are asked to automatically cleanup beacons with Robo which are randomly scattered on a large carpet. As an extra exercise the fastest students are asked to modify their program so that Robo will not collide with the wall anymore.
Week 7	**Repairing roads.** This is the this last lesson where student will do their 'masterpiece'. The goal is to mark/repair holes in the road by placing beacons on top of them. The beacons and holes are randomly distributed. To solve the problem two procedures have to be programmed for finding and placing the beacons. Random search is needed to do this correctly. In addition double conditions are necessary to check that a hole does not already have a beacon on top. When successful, students will be able to let Robo automatically mark all holes with beacons.

Performance monitoring

The lessons have an open training objective and are oriented towards problem solving. Main objective is to gain an understanding of logic, automation, robotics and how you can solve bigger problems by breaking these down into smaller sub problems. This insight is necessary to solve the problems in the exercises. The number of hints or clues a student needs to solve a problem gives some insight into how much understanding is acquired. At least as important, however, is that students enjoy the lessons, come to see that technology is really fun, and that the resolution of a problem can give a lot of satisfaction.

Tips

Let the children do as much as possible by themselves, but give clues if needed, such that they have a least seen a bit of all exercises.